High Order Orthogonal Tensor Networks: Information Capacity and Reliability

Alexander N. Gorban[¥], Yeugenii M. Mirkes[¥] and Donald C. Wunsch II[§]

[¥]Krasnoyarsk Computing Center RAS, Krasnoyarsk State Technical University
660036, Krasnoyarsk-36, Computing Center RAS, Russian Federation
E-mail: gorban@cc.krascience.rssi.ru

[§]Applied Computational Intelligence Laboratory
Department of Electrical Engineering, Texas Tech University
Lubbock, TX 79409-3102
E-mail: Dwunsch@coe2.coe.ttu.edu

Abstract

Neural networks based on construction of orthogonal projectors in the tensor power of space of signals are described. A sharp estimate of their ultimate information capacity is obtained. The number of stored prototype patterns (prototypes) can many times exceed the number of neurons. A comparison with the error control codes is made.

1. Introduction

The number of patterns which the Hopfield network can store and precisely reproduce has been reported to be limited to 14% of the number of neurons, and in this case the patterns must be weakly correlated. The most important is the condition of weak correlation, since in practice it is usually necessary to distinguish similar objects. For example, different letters of the alphabet are strongly correlated in most cases.

A large body of research has been made to modify the Hopfield networks to eliminate the restrictions mentioned. One modification is the projective network (see, for example, [2]). The main idea of the projective network is to turn the network connection matrix into an orthogonal projector. (A variety of other approaches have been analyzed by Michel, et. al. [7])

In contrast to the original Hopfield network [1], the projective network can distinguish strongly correlated patterns. However, if among the prototypes there exist N linearly independent vectors (where N is the number of neurons, i.e. dimension of the space), then the network matrix becomes an identity matrix, and the network transmits the input signals without any change.

The quadratic part of the "energy" H in the Hopfield networks is interpreted as an analogue of the potential energy of two-particle interaction. The transition to three-, four- and higher degrees of interaction gives rise to associative memory working much better than the Hopfield networks [3,4].

Similarly, in the case of projective network one can turn to the space of two-, three-, and of higher order k-particle functions [4,5]. The tensor networks proposed in the present paper combine the advantages of both projective and multi-particle networks.

2. High order orthogonal tensor networks (HOOT-networks)

Denote the set of prototypes as $\{x_i\}_{i=1}^m$. The tensor power $x^{\otimes k}$ is k-index variable $x^{\otimes k}_{i_1 i_2 \ldots i_k} = x_{i_1} \cdot x_{i_2} \cdot \ldots \cdot x_{i_k}$. The set of the vectors $\{v_i\}_{i=1}^m$ is called dual to the set of vectors $\{f_i\}_{i=1}^m$ if the following conditions are satisfied: $(f_i, v_i) = 1$, $i = 1, \ldots, m$; $(f_i, v_j) = 0$ under $i \neq j$ and $\{v_i\}_{i=1}^m$ belongs to the linear envelope of $\{f_i\}_{i=1}^m$. If the set of vectors $\{f_i\}_{i=1}^m$ is linearly independent, then the vectors of the dual set are calculated from the formula

$$v_i = \sum_{j=1}^{m} (g^{-1})_{ij} f_j , \qquad (1)$$

0-7803-4122-8/97 $10.00©1997 IEEE

where $(g^{-1})_{ij}$ is the *ij*-th element of the matrix which is inverse to the Gram matrix of the set of vectors $\{f_i\}_{i=1}^{m}$, with the elements $g_{ij} = (f_i, f_j)$.

Let the coordinates of the vectors be only ±1. The tensor network of the valence *k* transforms the input vector *x* into output vector *x'* in the following way:

$$x' = \mathrm{Sign}\left(\sum_{i=1}^{m}\left(v_i, x^{\otimes k}\right)x_i\right), \qquad (2)$$

where $\{v_i\}_{i=1}^{m}$ is the set of vectors, dual to the set $\{x_i^{\otimes k}\}_{i=1}^{m}$; @ Sign is a coordinate-wise acting function defined by the following formula: $(\mathrm{Sign}(y))_i = \mathrm{sign}(y_i)$. Since $\left(a^{\otimes k}, b^{\otimes k}\right) = (a, b)^k$, one can rewrite (2) in the following form:

$$x' = \mathrm{Sign}\left(\sum_{i=1}^{m}\sum_{j=1}^{m}(g^{-1})_{ij}\left(x_j, x\right)^k x_i\right), \qquad (3)$$

where $(g^{-1})_{ij}$ is the *ij*-th element of the matrix which is inverse to the Gram matrix of the set of vectors $\{x_i^{\otimes k}\}_{i=1}^{m}$ with the elements $g_{ij} = (x_i, x_j)^k$.

Formula (3) does not use the tensors, thus the calculation time and the memory required for this do not depend on the valence of the tensors.

3. Information capacity of HOOT-networks

The benefit of transition to projecting in the tensor space can be briefly explained as follows: tensor powering can turn the linearly dependent vectors of prototypes, thus increasing the information capacity of the network.

The memory of the projection network is "absolute": when one of the prototypes is given to the input of network this prototype is also at the output. This merit of the network is useless when the number of prototypes exceeds a certain value and the network becomes "transparent" - every input vector yields the same output vector. Sometimes "transparency" can be eliminated by increasing valence.

The information capacity of the tensor network of *k* valence is considered to mean the number of prototypes which the tensor network of this valence is capable to remember and reproduce without errors. The question of the upper bound of the information capacity is reduced to the question of maximum possible rank of the vectors' set $\{x_i^{\otimes k}\}_{i=1}^{m}$

The simplest but very excessive estimate is given by the value n^k. To be more exact the rank sought for does not exceed the dimension of the symmetrical tensor space. This dimension is found by the Euler formula and is equal to $\mathbf{C}_{n+k-1}^{k-1}$ (where $\mathbf{C}_m^l$ is the binomial coefficient of *m* by *l*). Yet, even this estimate is excessive.

Theorem. With $k < n$

$$\max\left\{\mathrm{rank}\left\{x^{\otimes k}\right\}\right\} = \sum_{i=0}^{k}\mathbf{C}_{n-1}^{i}.$$

Denote this value r_{nk}. A small modification of Pascal's triangle (Fig. 1) is used to calculate r_{nk}. The first line contains two, since with *n=2* there always are two non-collinear vectors in the set. In the transition to the next line the first element is produced by adding a unity to the first element of the previous line, the second - as the sum of the first and second elements of the previous line, the third - as the sum of the second and the third elements, and so on.

The last element is produced by doubling the last element of the previous line.

Table 1 compares three estimates for certain values of *n* and *k*. One can easily see that the correction in transition to the estimate r_{nk} is quite considerable. The limit information capacity can, on the other hand, considerably exceed the number of neurons.

4. Reliability of HOOT-networks

It is important to find out how reliable is the operation of the neural network in the presence of noise, and how often it correctly transforms the input vector into the nearest prototype. The operation of tensor networks in the presence of noise was compared to the potentialities of the linear codes correcting errors. By a linear code correcting *k* errors we call a linear subspace in the *n*-dimensional space over GF_2 all vectors of which are distant from each other not less than by $2k+1$ (see, for example [6]). A linear code is called perfect when for every vector of the *n*-dimensional space there is a cod e vector distant from the given one by not more than *k*. The tensor network input was given all code vectors of the code taken for the sake of comparison.

Numerical experiment with perfect codes demonstrated that the tensor network of the minimum required valence decodes all vectors correctly. For the imperfect codes the picture was worse - among the stable images there were spurious states - vectors that did not belong to the set of prototypes.

Detailed results of experiments are given in Tables 2 and 3. In the case of n=10, k=1 (see Tables 2 and 3, line 1) with valences 3 and 5 the tensor network operated as an identical operator - all input vectors were transmitted to the network output without changes. With the valence of k = 7 the number of spurious states drastically dropped and the network decoded more than 60% of the signal correctly. At this, all vectors distant from the nearest prototype by distance of 2 were decoded correctly, while the part of vectors distant from the nearest prototype by the distance of 1, remained spurious states . In the case of n=10, k=2 (see Tables 2 and 3, lines 3, 4, 5) the number of spurious states was observed to decrease with increasing valence. The network correctly decoded more than 50% of signals. However, even at n=15, k=3 and the valence more than 3 (see Tables 2 and 3, lines 6, 7) the network decoded all signals with three errors correctly. In most experiments the number of prototypes exceeded the number of neurons.

So, the quality of operation of the tensor network increases with dimension of space and valence, while in terms of eliminating errors it approaches the error-correcting code.

References

1. Hopfield J.J. *Neural Networks and physical systems with emergent collective computational abilities.* Proc. Nat. Sci. USA. 1982. V.79. pp. 2554-2558.
2. Gorban A.N. *Training neural networks*, USSR-USA JV "ParaGraph", Moscow, 1990 (Russian). (English Translation: AMSE Transaction, Scientific Siberian, A, vol.6. Neurocomputing, pp.1-134, 1993).
3. Konopljov V.A., Sinitzyn Ye.V. *Neural networks with maximal information capacity.* III All-Russian Workshop "Neuroinformathiks and Applications", Book of abstracts. Krasnoyarsk Technical University Press, 1995, p. 66.
4. Gorban A.N., Rossiev D.A. *Neural Networks on PC.* Novosibirsk: Nauka Publ., 1996.
5. Golub D.N., Gorban A.N. *Multi-Particle Networks for Associative Memory.* Proc. of the World Congress on Neural Networks (WCNN'96), Sept. 15-18, 1996, San Diego, CA, Lawrence Erlbaum Associates, 1996, pp. 772-775.
6. Blahut R.E. *Theory and practice of error control codes.* Addison-Wesley, 1984.
7. Michel, Anthony, *Qualitative Limitations Incurred in Implementations of Recurrent Neural Networks*, IEEE Control Systems Magazine, June 1995, pp. 52- 65. This survey references a variety of Michel's and collaborators' earlier works.

2
3 4
4 7 8
5 11 15 16
6 16 26 31 32
7 22 42 57 63 64
8 29 64 99 120 127 128
9 37 93 163 219 247 255 256
10 46 130 256 382 466 502 511 512
$r_{11,1}$ $r_{11,2}$ $r_{11,3}$ $r_{11,4}$ $r_{11,5}$ $r_{11,6}$ $r_{11,7}$ $r_{11,8}$ $r_{11,9}$ $r_{11,10}$

Fig. 1. Modified Pascal triangle

Table 1

n	k	n^k	C_{n+k-1}^{k-1}	$r_{n,k}$
5	2	25	15	11
	3	125	35	15
10	3	1 000	220	130
	6	1 000 000	5005	466
	8	100 000 000	24310	511

Table 2
Numerical Experiment with HOOT-Networks
(Network Effect on All Input ± 1 Vectors of Given Dimension)

1	Dimension of signals	Number of ±1 vectors	Minimal distance between prototypes	Number of prototypes	Tensor degree	Number of spurious states	Number of network answers:		Number of input vectors for which the distance from the true prototypes after passage through network is:		
							True	False	<(decr.)	=(conserv.)	>(incr.)
1	10	1024	3	64	3÷5	896	128	896	0	856	0
2					7÷21	384	640	384	0	348	0
3	10	1024	5	8	3	260	464	560	240	260	60
4					5÷15	230	494	530	240	230	60
5					17÷21	140	532	492	240	182	70
6	15	32768	7	32	3	15456	17312	15456	0	15465	0
7					5÷21	14336	18432	14336	0	14336	0

Table 3
Distribution of Errors and spurious states by Distance to the Nearest Prototype
(Experiment Numbers as in Table 2)

1	Number of spurious states with distance *d* from nearest prototype:					Number of input vectors for which the network answer is false with distance *d* from nearest prototype:				
	d=1	*d*=2	*d*=3	*d*=4	*d*=5	*d*=1	*d*=2	*d*=3	*d*=4	*d*=5
1	640	256	0	0	0	896	0	0	0	0
2	384	0	0	0	0	384	0	0	0	0
3	0	210	50	0	0	0	210	290	60	0
4	0	180	50	0	0	0	180	290	60	0
5	0	88	50	2	0	0	156	290	60	0
6	0	0	1120	13440	896	0	0	1120	13440	896
7	0	0	0	13440	896	0	0	0	13440	896